C(PROGRAMMING : UDF)

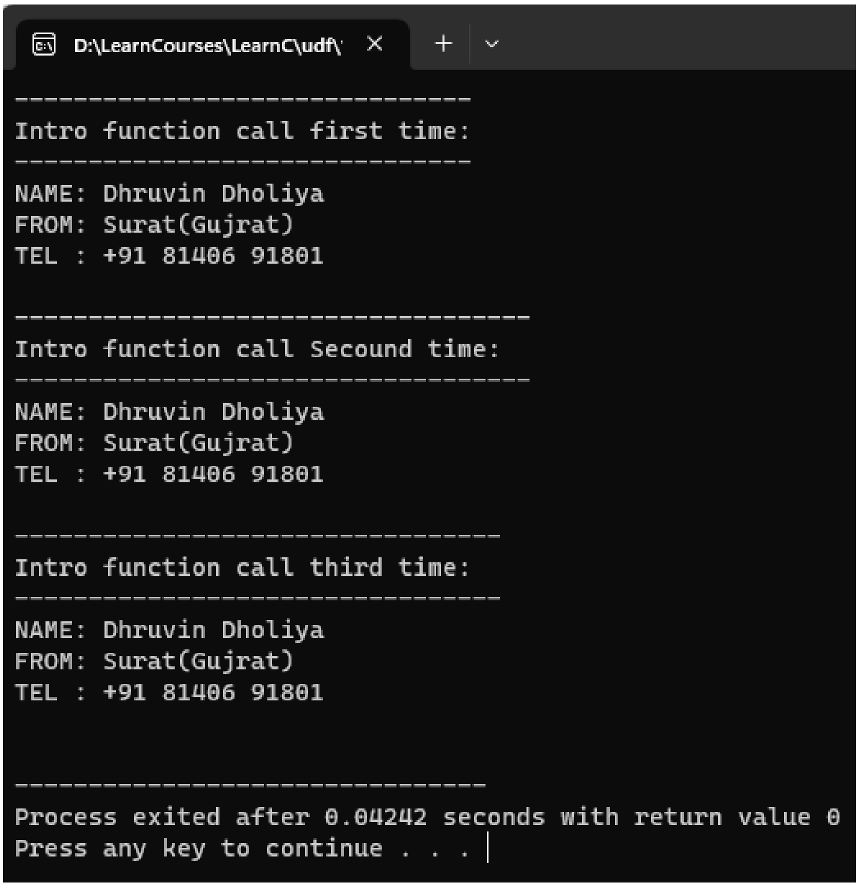
Dhruvin Dholiya

1. Write a C program to print your introduction using type-1 function.

❖ Input:

```
// ====== no parameters, no return value. =======

#include<stdio.h>

void intro() {

      printf("NAME: Dhruvin Dholiya\n");
      printf("FROM: Surat(Gujrat)\n");
      printf("TEL : +91 81406 91801\n\n");

}

int main() {
      printf("------------------------------\n");
      printf("Intro function call first time:\n");
      printf("------------------------------\n");

      intro();

      printf("----------------------------------");
      printf("\nIntro function call Secound time:\n");
      printf("----------------------------------\n");

      intro();

      printf("--------------------------------");
      printf("\nIntro function call third time:\n");
      printf("--------------------------------\n");

      intro();

      return 0;
}
```

❖ Output:



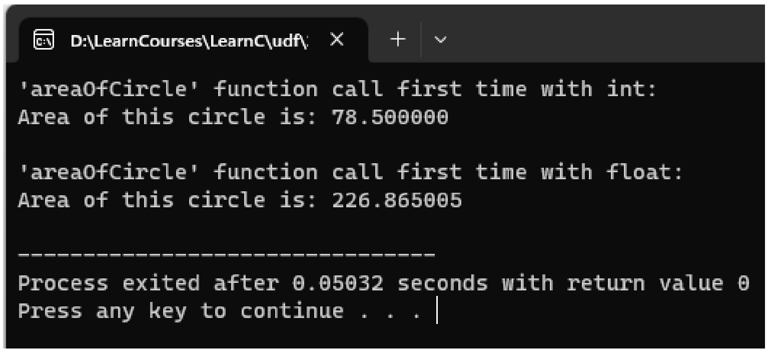
```
-------------------------------
Intro function call first time:
-------------------------------
NAME: Dhruvin Dholiya
FROM: Surat(Gujrat)
TEL : +91 81406 91801

-----------------------------------
Intro function call Secound time:
-----------------------------------
NAME: Dhruvin Dholiya
FROM: Surat(Gujrat)
TEL : +91 81406 91801

---------------------------------
Intro function call third time:
---------------------------------
NAME: Dhruvin Dholiya
FROM: Surat(Gujrat)
TEL : +91 81406 91801



--------------------------------
Process exited after 0.04242 seconds with return value 0
Press any key to continue . . .
```

2. Write a C program to find area of circle using type-2 function.

❖ Input:

```
//========== with parameters, no return value.========

#include<stdio.h>

void areaOfCircle(float r) {

    float area;

    area = 3.14 * r * r;

    printf("Area of this circle is: %f\n", area);

}

int main() {

    printf("'areaOfCircle' function call first time with int:\n");

    areaOfCircle(5);

    printf("\n'areaOfCircle' function call first time with float:\n");

    areaOfCircle(8.5);

    return 0;
}
```

❖ Output:



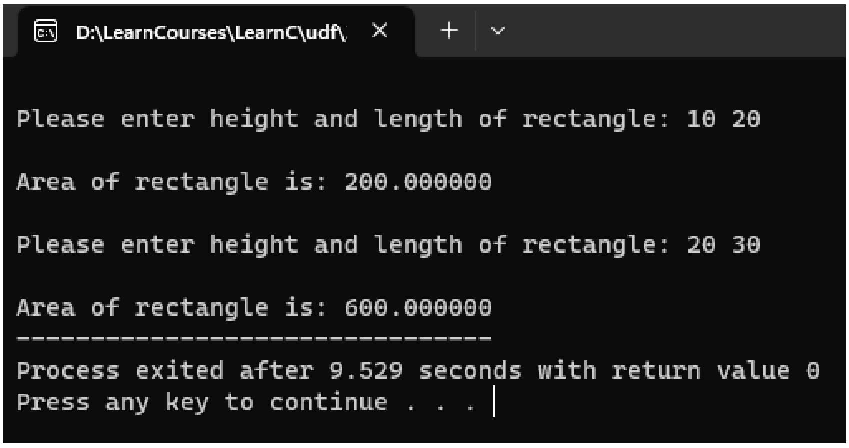
```
'areaOfCircle' function call first time with int:
Area of this circle is: 78.500000

'areaOfCircle' function call first time with float:
Area of this circle is: 226.865005

--------------------------------
Process exited after 0.05032 seconds with return value 0
Press any key to continue . . .
```

3. Write a C program to find area of rectangle using type-3 function.

❖ Input:

```
// =========== no parameters, with return value. ===========
#include<stdio.h>

float areaOfrect() {
    float area, h, l;

    printf("\nPlease enter height and length of rectangle: ");
    scanf("%f %f", &h, &l);

    area = h * l;

    return area;
}

int main() {

    float res, res1;

    res = areaOfrect();
    printf("\nArea of rectangle is: %f\n", res);

    res1 = areaOfrect();
    printf("\nArea of rectangle is: %f", res1);

    return 0;
}
```

❖ Output:



```

Please enter height and length of rectangle: 10 20

Area of rectangle is: 200.000000

Please enter height and length of rectangle: 20 30

Area of rectangle is: 600.000000
--------------------------------
Process exited after 9.529 seconds with return value 0
Press any key to continue . . .
```

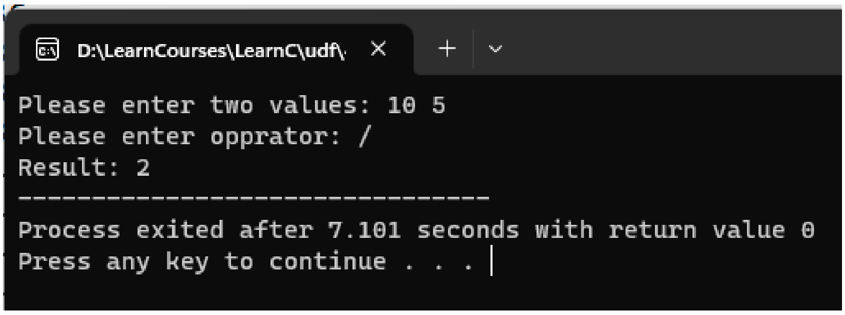
4. Write a C program to create a calculator using type-4 function.

❖ Input:

```
// ===========  with parameters, with return value. =========

#include<stdio.h>

int add(int a, int b) {
      int val;

      val = a + b;

      return val;
}

int sub(int a, int b) {
      int val;

      val = a - b;

      return val;
}

int mul(int a, int b) {
      int val;

      val = a * b;

      return val;
}

int div(int a, int b) {
      int val;

      val = a / b;

      return val;
}

int main() {

      int res, a, b;
      char op;

      printf("Please enter two values: ");
      scanf("%d %d", &a, &b);
```

```
        printf("Please enter opprator: ");
        scanf(" %c", &op);

        switch(op) {
                case '+':
                        res = add(a, b);
                        break;
                case '-':
                        res = sub(a, b);
                        break;
                case '*':
                        res = mul(a, b);
                        break;
                case '/':
                        res = div(a, b);
                        break;
                default:
                        printf("\n\nINVALID - Please check your added input.\n\n");
                        break;
        }

        printf("Result: %d", res);


        return 0;
}
```

❖ Output:



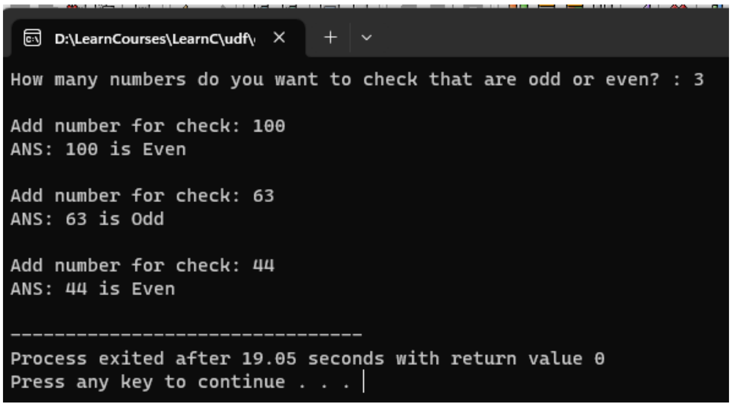
```
Please enter two values: 10 5
Please enter opprator: /
Result: 2
--------------------------------
Process exited after 7.101 seconds with return value 0
Press any key to continue . . .
```

5. Write a C program to find number is even or odd using type-1 function.

❖ Input:

```
#include <stdio.h>
void evenOdd() {
   int i, n, num, res;

        printf("How many numbers do you want to check that are odd or
even? : ");
   scanf("%d", &n);

        for(i = 1; i <= n; i++) {

           printf("\nAdd number for check: ");
           scanf("%d", &num);

        res = num % 2;

                if (res == 0) {
                printf("ANS: %d is Even\n", num);
                } else {
                        printf("ANS: %d is Odd\n", num);
                }
        }
}

int main() {

        evenOdd();

        return 0;
}
```

❖ Output:



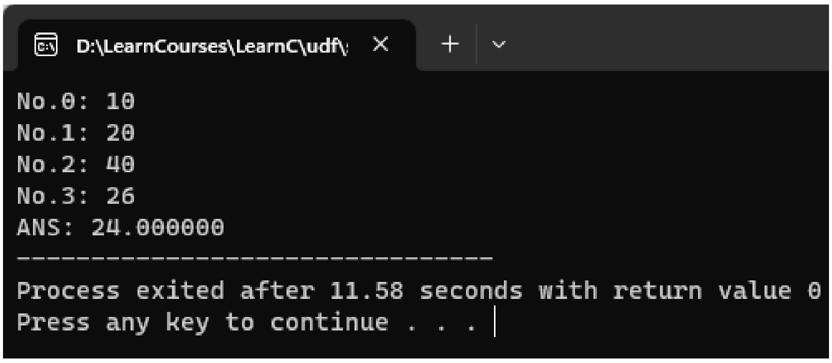
```
How many numbers do you want to check that are odd or even? : 3

Add number for check: 100
ANS: 100 is Even

Add number for check: 63
ANS: 63 is Odd

Add number for check: 44
ANS: 44 is Even

--------------------------------
Process exited after 19.05 seconds with return value 0
Press any key to continue . . .
```

6. Write a C program to find average of 4 numbers using type-2 function.

❖ **Input:**

```
#include<stdio.h>

void average(int arr[]) {
      int i;
      float sum = 0, avg = 0;

      for(i = 0; i < 4; i++) {
                sum = (sum + arr[i]);
      }
      avg = sum/4;
      printf("ANS: %f", avg);

}

int main() {

      int arr[4], n, i;

      for (i = 0; i < 4; i++) {
                printf("No.%d: ", i);
                scanf("%d", &arr[i]);
      }

      average(arr);

      return 0;
}
```

❖ **Output:**



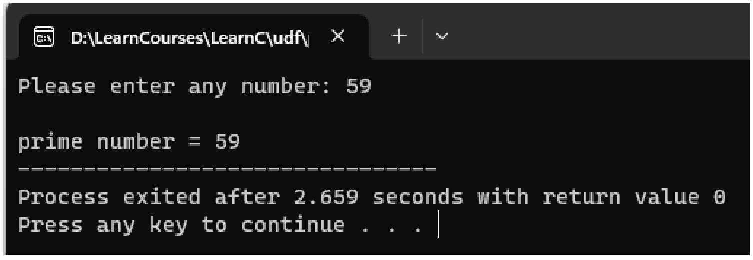
```
No.0: 10
No.1: 20
No.2: 40
No.3: 26
ANS: 24.000000
--------------------------------
Process exited after 11.58 seconds with return value 0
Press any key to continue . . .
```

7. Write a C program to find given number is prime or not using type-3 function.

❖ Input:

```
#include<stdio.h>
int primeNum() {
    int num, i, prime = 0;
    printf("Please enter any number: ");
    scanf("%d", &num);

    if (num > 1) {
            if (num == 2) {
                    printf("prime number = ");
            } else {
                    for (i = 2; i < num; i++) {
                            if (num % i == 0) {
                                    prime = 1;
                                    break;
                            }
                    }
                    if (prime == 0) {
                            printf("\nprime number = ");
                    } else {
                            printf("\nnot prime number = ");
                    }
            }
    } else {
            printf("Added number is should more than 1.");
    }
    return num;
}
int main() {

    int res = primeNum();
    printf("%d", res);
    return 0;
}
```

❖ Output:



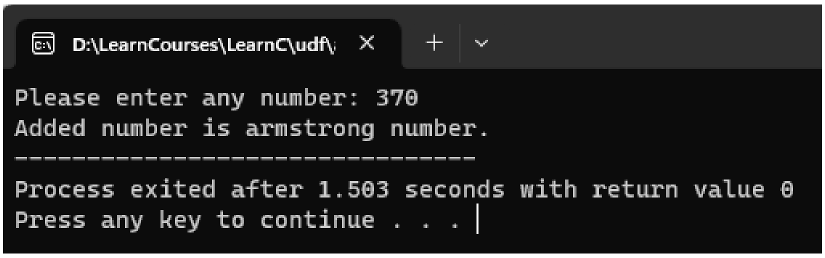
```
Please enter any number: 59

prime number = 59
--------------------------------
Process exited after 2.659 seconds with return value 0
Press any key to continue . . .
```

8. Write a C program to find given number is Armstrong or not using type-4 function.

❖ **Input:**

```
#include<stdio.h>

int armstrongNum(int num){

    int i, rem=0, res=0, originalNum;

    originalNum = num;

    for (i = 0; i < num; i++) {
            rem = num % 10;
            res = res + (rem * rem * rem);
            num = num / 10;
    }

    if (originalNum == res) {
            printf("Added number is armstrong number.");
    } else {
            printf("Added number is not armstrong number.");
    }
    return res;
}

int main () {
    int num;

    printf("Please enter any number: ");
    scanf("%d", &num);

    armstrongNum(num);

    return 0;
}
```

❖ **Output:**



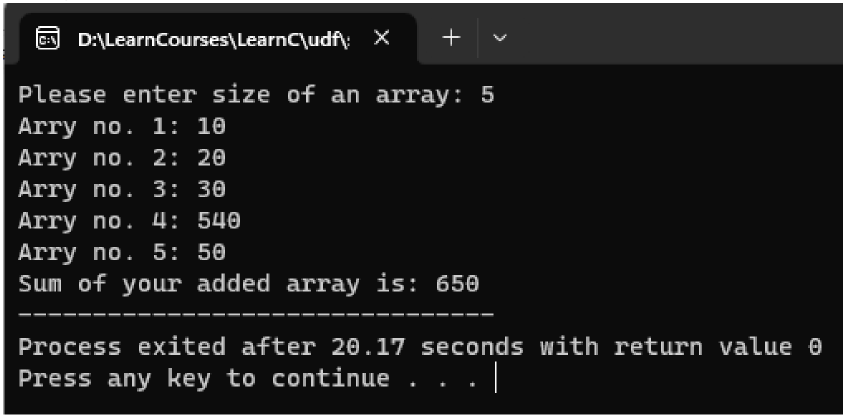
```
Please enter any number: 370
Added number is armstrong number.
--------------------------------
Process exited after 1.503 seconds with return value 0
Press any key to continue . . .
```

9. C program to find Sum of all Array Elements by passing array as an argument using User Define Functions.

❖ **Input:**

```
#include <stdio.h>

int sumArr(int arr[], int n) {

    int sum = 0, i;

    for (i=0; i<=n; i++) {
            sum = sum + arr[i];
    }

    return sum;
}

int main() {

    int n, i, arr[100], res;

    printf("Please enter size of an array: ");
    scanf("%d", &n);

    for (i=0; i < n; i++) {
            printf("Arry no. %d: ", i + 1);
            scanf("%d", &arr[i]);
    }

    res = sumArr(arr, n);
    printf("Sum of your added array is: %d", res);

    return 0;
}
```

❖ **Output:**



```
Please enter size of an array: 5
Arry no. 1: 10
Arry no. 2: 20
Arry no. 3: 30
Arry no. 4: 540
Arry no. 5: 50
Sum of your added array is: 650
--------------------------------
Process exited after 20.17 seconds with return value 0
Press any key to continue . . .
```

10. C program to find Length of the String by passing String/ Character Array as an Argument using User Define Functions.

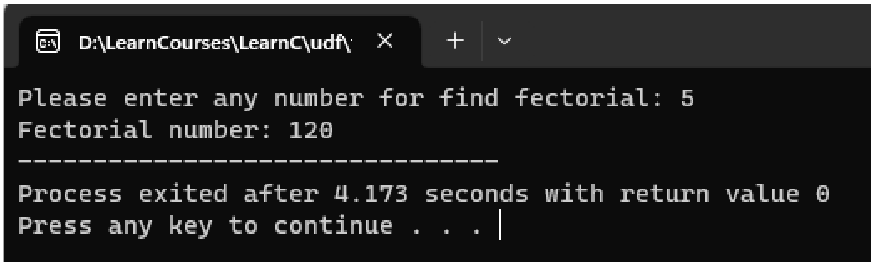
- **Input:**
- **Output:**

11.C program to find factorial of number using recursion.

- **Input:**

```
#include <stdio.h>

int findFectNum(int num) {

        if (num > 1) {
                return num * findFectNum(num - 1);
        } else {
                return 1;
        }

}

int main() {

        int num, i, res;

        printf("Please enter any number for find fectorial: ");
        scanf("%d", &num);

        res = findFectNum(num);
        printf("Fectorial number: %d", res);

        return 0;
}
```

- **Output:**



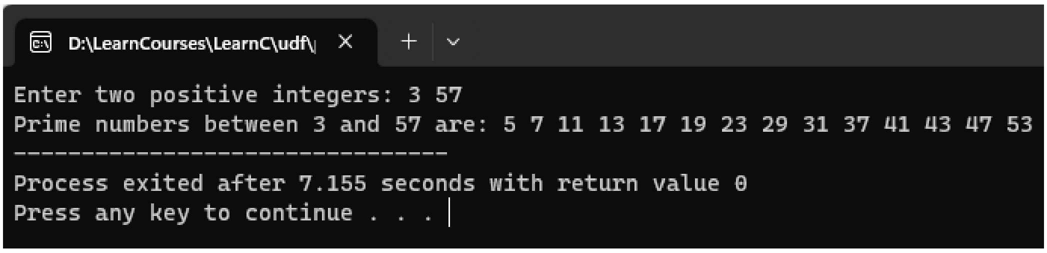
```
Please enter any number for find fectorial: 5
Fectorial number: 120
--------------------------------
Process exited after 4.173 seconds with return value 0
Press any key to continue . . .
```

12. Write C program to print prime number of between two any number.

❖ **Output:**

```
#include <stdio.h>


int checkPrimeNumber(int i) {
  int j, flag = 1;
  for (j = 2; j <= i / 2; j++) {
    if (i % j == 0) {
      flag = 0;
      break;
    }
  }
  return flag;
}
int main() {
  int i, flag, arr[2];
  printf("Enter two positive integers: ");
  scanf("%d %d", &arr[0], &arr[1]);

  printf("Prime numbers between %d and %d are: ", arr[0], arr[1]);
  for (i = arr[0] + 1; i < arr[1]; i++) {
    flag = checkPrimeNumber(i);
    if (flag == 1) {
      printf("%d ", i);
    }
  }
  return 0;
}
```

❖ Output:



```
Enter two positive integers: 3 57
Prime numbers between 3 and 57 are: 5 7 11 13 17 19 23 29 31 37 41 43 47 53
--------------------------------
Process exited after 7.155 seconds with return value 0
Press any key to continue . . .
```

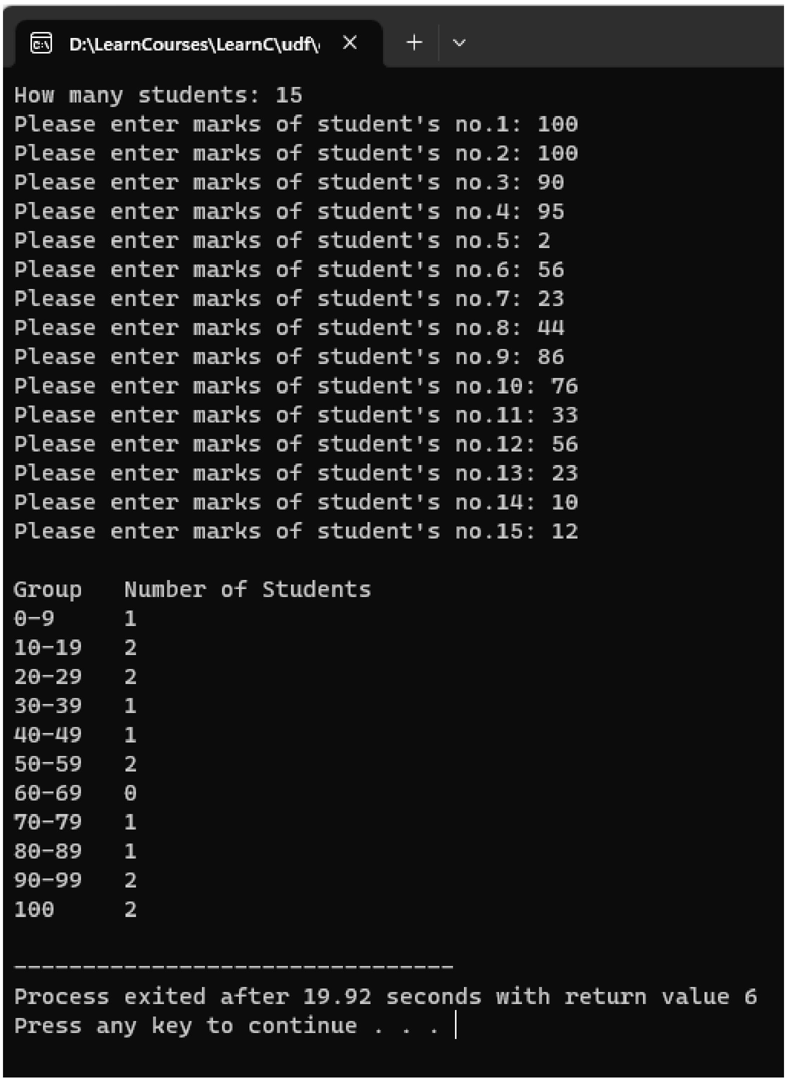
13.Write a C program to count number of students in each group (0-9, 10- 19, 20-29 .... 90-99, 100-100) for the given students marks by funcation and array.

Marks: 85, 66, 37, 45, 68, 23, 99, 100, 81, 70, 42, 55, 68, 77, 96, 18

❖ **Output:**

```
#include <stdio.h>
void countStudent(int n) {
  int marks[100], i, group, count[11] = {0};
        for (i = 0; i < n; i++) {
                printf("Please enter marks of student's no.%d: ", i + 1);
                scanf("%d", &marks[i]);
        }
  for (i=0; i<n; i++) {
    group = marks[i] / 10;
    count[group]++;
  }
  printf("\nGroup\tNumber of Students\n");
  for (i=0; i<10; i++) {
    printf("%d-%d\t%d\n", i * 10, i * 10 + 9, count[i]);
  }
    printf("100\t%d\n", count[10]);
}

int main () {
        int n;
        printf("How many students: ");
        scanf("%d", &n);

        countStudent(n);
}
```

❖ **Output:**



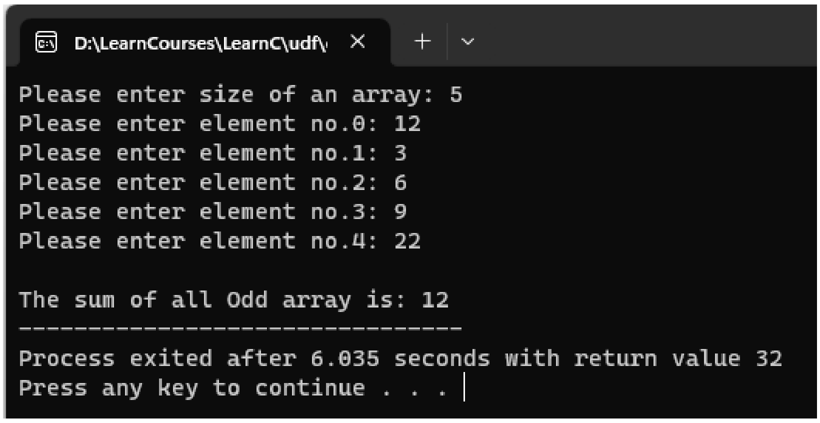
```
How many students: 15
Please enter marks of student's no.1: 100
Please enter marks of student's no.2: 100
Please enter marks of student's no.3: 90
Please enter marks of student's no.4: 95
Please enter marks of student's no.5: 2
Please enter marks of student's no.6: 56
Please enter marks of student's no.7: 23
Please enter marks of student's no.8: 44
Please enter marks of student's no.9: 86
Please enter marks of student's no.10: 76
Please enter marks of student's no.11: 33
Please enter marks of student's no.12: 56
Please enter marks of student's no.13: 23
Please enter marks of student's no.14: 10
Please enter marks of student's no.15: 12

Group   Number of Students
0-9     1
10-19   2
20-29   2
30-39   1
40-49   1
50-59   2
60-69   0
70-79   1
80-89   1
90-99   2
100     2

--------------------------------
Process exited after 19.92 seconds with return value 6
Press any key to continue . . .
```

14.Write C program to calculate sum of n odd elements.

- **Output:**

```
#include <stdio.h>

int oddSum() {
        int i, n, arr[i], sum=0;

        printf("Please enter size of an array: ");
        scanf("%d", &n);

        for (i = 0; i < n; i++) {
                printf("Please enter element no.%d: ", i);
                scanf("%d", &arr[i]);
        }

        for(i = 0; i < n; i++) {
                if(arr[i] % 2 == 1) {
                        sum = sum + arr[i];
                }

        }



        return sum;
}
int main() {
        int sum;

        sum = oddSum();
        printf("\nThe sum of all Odd array is: %d", sum);
    }
```

❖ Output:



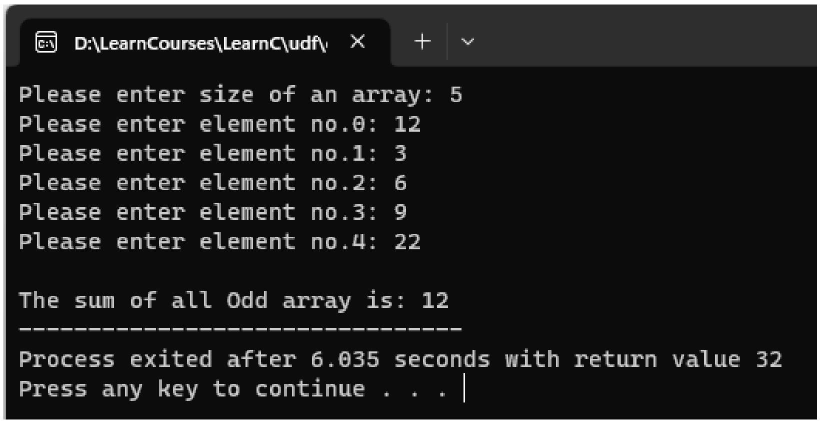
```
Please enter size of an array: 5
Please enter element no.0: 12
Please enter element no.1: 3
Please enter element no.2: 6
Please enter element no.3: 9
Please enter element no.4: 22

The sum of all Odd array is: 12
--------------------------------
Process exited after 6.035 seconds with return value 32
Press any key to continue . . .
```

15.Write C program to print smallest element of array.

❖ Output:

```
#include <stdio.h>
void smallestNum() {
        int i, n, arr[i], max=0;
        printf("Please enter size of an array: ");
        scanf("%d", &n);
        for (i = 0; i < n; i++) {
        printf("Array element no.%d: ", i);
                scanf("%d", &arr[i]);
        }
        for (i = 0; i < n; i++) {
                if (arr[i] < max) {
                        max = arr[i];
                }
        }
        printf("\nmax: %d", max);
}
int main() {
        smallestNum();
}
```

❖ Output:



```
Please enter size of an array: 5
Please enter element no.0: 12
Please enter element no.1: 3
Please enter element no.2: 6
Please enter element no.3: 9
Please enter element no.4: 22

The sum of all Odd array is: 12
--------------------------------
Process exited after 6.035 seconds with return value 32
Press any key to continue . . .
```